International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 2840-2853

https://doi.org/10.70517/ijhsa464236

Model construction based on principal component analysis to optimize the accuracy of financial market price volatility prediction

Minmin Huo[1,*]
[1] Sichuan Vocational and Technical College of Communications, Chengdu, Sichuan, 611130, China
Corresponding authors: (e-mail: huominmin0112@163.com).

Abstract Optimizing the prediction of financial market price fluctuations and constructing more effective financial market price prediction models has always been a research topic of great interest to both the academic and practical communities in the field of financial markets. To this end, this paper combines BP neural network technology with principal component analysis (PCA) to construct a stock price prediction model based on PCA-BP neural networks. This paper selects the CSI 300 Index as the research object and conducts empirical analysis using the stock price prediction model constructed in this paper. The results show that the model has the highest prediction accuracy. In the directional accuracy analysis on December 5, the accuracy rate reached 92.63%, which can provide decision-making basis for investors and regulators to a certain extent.

Index Terms BP neural network, principal component analysis, PCA-BP neural network, stock price prediction model

I. Introduction

Financial markets are complex systems involving numerous participants and large amounts of capital, and price fluctuations are one of their inherent characteristics. External factors such as changes in global economic conditions, political instability, natural disasters, and the use of technological tools, as well as internal factors such as shifts in market supply and demand, corporate financial health, and market participant sentiment, can all trigger fluctuations in market sentiment, thereby influencing price volatility in financial markets. These factors are characterized by uncertainty and suddenness, making price volatility prediction complex and challenging [1]-[5]. Predicting market price fluctuations is a common concern for investors, traders, financial professionals, and policymakers. By establishing effective predictive models, one can better understand market dynamics and make more informed investment decisions [6], [7]. Price fluctuations in financial markets exhibit randomness and nonlinearity, and the quality and completeness of financial market data also influence price fluctuation predictions. The absence of historical data or the presence of data anomalies can introduce biases in prediction results, making predictions highly challenging [8]-[10]. Therefore, improving the accuracy of financial market price fluctuation predictions has become one of the hot topics in this field.

Based on existing literature, financial market price volatility prediction can be categorized into traditional prediction methods, machine learning methods, and deep learning methods. Traditional prediction methods primarily refer to econometric methods (including the Autoregressive Integrated Moving Average (ARIMA) model, the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, etc.). Jiang and Subramanian et al. [11] utilized the ARIMA model, which excels in analyzing time-series data, to predict future stock prices, achieving satisfactory results. Lin [12] validated the fitting and predictive performance of GARCH, threshold GARCH, and exponential GARCH models on stock price volatility, with the exponential GARCH model performing the best. Alshammari et al. [13] constructed a stock market price volatility prediction model combining the maximum overlap discrete wavelet transform function and the exponential GARCH model, which demonstrated higher accuracy compared to nonlinear spectral models, ARIMA models, and exponential GARCH models. In terms of exchange rate volatility prediction, Abdullah et al. [14] compared the predictive performance of four models—GARCH, asymmetric power ARCH, exponential GARCH, and integer GARCH—under normal distribution and Student's t-distribution. Student's t-distribution improved predictive performance, with GARCH yielding the best results. However, these methods have limitations in fitting the non-stationary and non-linear characteristics of financial data and interpreting high-dimensional data.

With the development of artificial intelligence, machine learning methods can fit certain nonlinear relationships in

data, achieving prediction accuracy for financial time series that far exceeds that of traditional models such as ARIMA and GARCH. These machine learning models primarily include support vector machines (SVM), backpropagation neural networks (BPNN), and multi-layer perceptrons (MLP). Özorhan et al. [15] utilized SVM to predict foreign exchange market exchange rate fluctuations under a genetic algorithm processing related currency price input data and a trading strategy involving currency strength. The predictive accuracy under the trading strategy was further enhanced. Nadh and Prasad [16] argued that SVM is the preferred choice among various machine learning methods for currency market predictions, and the predictive accuracy was improved by integrating SVM with genetic algorithms. Due to SVM's limitations in handling imbalanced data, Mohammed [17] developed an SVM stock price prediction model based on oversampling methods and cost-sensitive learning mechanisms, achieving higher prediction accuracy and reliability. Zhang and Lou et al. [18] used BPNN and deep learning fuzzy algorithms to predict stock price fluctuations, achieving accuracies of 73.29% and 62.12%, respectively. Alameer et al. [19] used the whale optimization algorithm as a trainer to learn the MLP neural network, establishing a hybrid model for predicting gold price fluctuations. Compared to the ARIMA model and the genetic algorithm-optimized MLP neural network, the prediction accuracy was improved by 85.84% and 25.40%, respectively. Vrtagic and Dogan [20] used genetic algorithms to optimize MLP to extract high-frequency data correlations and constructed a gold price prediction model, achieving an accuracy of over 99%. However, financial data influenced by economic, political, and public opinion factors has high-dimensional characteristics. Shallow machine learning algorithms such as SVM and BPNN face issues like the curse of dimensionality and ineffective feature representation when learning high-dimensional data.

In recent years, with the significant improvement in computing power and the maturation of deep learning methods, deep learning models have demonstrated stronger feature extraction capabilities and high adaptability compared to traditional neural networks, enabling them to fit the nonlinear characteristics of financial data and further enhance prediction accuracy [21], [22]. These deep learning models primarily include recurrent neural networks (RNN), long short-term memory networks (LSTM), and convolutional neural networks (CNN). Pan et al. [23] combined LSTM and GARCH to predict stock market prices, not only addressing the interference caused by the non-stationary and nonlinear characteristics of prices but also significantly enhancing prediction accuracy. Zhang et al. [24] utilized wavelet transforms to process price data fluctuations and high-frequency noise interference when predicting Bitcoin and gold prices, and employed an optimized LSTM model for prediction, thereby improving prediction accuracy and precision. Dey et al. [25] compared the predictive performance of RNN, LSTM, and gated recurrent units (GRU) in stock price volatility. The vanishing gradient effect caused the basic RNN to perform the worst, while GRU had smaller errors compared to LSTM. Additionally, the reliability of all three models improved as the time intervals between stock prices decreased. From the above analysis, most financial time series trend prediction studies based on deep learning methods use one-dimensional time series input models for prediction. However, financial market price fluctuations are strongly influenced by various complex factors, exhibiting inherent high-dimensional characteristics. Additionally, data missing issues reduce the model's ability to process data [26], [27].

Furthermore, to ensure the effectiveness of predictions, non-stationary financial time series heavily rely on prior knowledge for data processing quality. Converting time series into images can help overcome these issues, as images can represent information across multiple dimensions of data, adapt to different data distributions and changes, and facilitate the model's feature extraction process [28], [29]. Wu et al. [30] proposed a novel graph-based CNN model for stock price prediction, which improved the accuracy of movement prediction, offering an effective method for achieving higher predictive accuracy in stock trading. However, the high-dimensional data processing performance and computational efficiency of these methods remain major challenges in current financial market price volatility prediction.

Principal Component Analysis (PCA) is a widely used statistical method for data dimensionality reduction and feature extraction. It transforms multiple variables in the original dataset into a few mutually independent principal components through orthogonal transformation. These principal components maximize the retention of information from the original dataset while maintaining efficient computational performance [31], [32]. Manoj and Suresh [33] used PCA to explore multicollinearity in financial variables, designing a multiple linear regression model for predicting gold prices, thereby enhancing the model's performance. Zhang, Z [34] designed a PCA-LSTM model, using PCA to reduce the dimensionality of stock price-related data, significantly reducing the poor generalization ability and prediction performance caused by data variables. Zhang, H [35] combined principal component analysis and BPNN to create a stock price prediction model, obtaining a stable and accurate prediction model through dimensionality reduction processing and a comparative analysis of mean squared error and mean absolute error.

The study first outlines the challenges in current financial market forecasting. Based on this, it provides a theoretical introduction to BP neural networks and principal component analysis. By combining principal component analysis with the BP neural network model, a PCA-BP neural network prediction model is constructed. Principal

component analysis reduces the number of input variables through dimensionality reduction, simplifying the neural network structure and enhancing the efficiency and accuracy of the BP neural network. Subsequently, the CSI 300 Index was selected as the research object, with a total of 4,000 data sets from March 1, 2010, to December 26, 2024, divided into training and testing groups. The model proposed in this paper was used for stock data analysis. Simulations were also conducted using the data to evaluate the effectiveness of the predictive model.

II. Financial market price prediction model

II. A. Challenges in Financial Market Forecasting

The stock market is a dynamic, nonlinear system influenced by various factors, and stock prediction faces numerous challenges that remain to be addressed, primarily in the following areas:

(1) Stock variables contain a significant amount of noise

Financial markets are large and complex dynamical systems with numerous internal and external factors influencing stock prices. Stock price time series data often contain a significant amount of noise. Additionally, there are countless technical indicators used for stock index prediction in the securities market. In the complex environment of the stock market, the more technical indicators selected subjectively for a specific stock or financial instrument, the more noise interferes with predictions, thereby limiting the speed and accuracy of stock price prediction models. Therefore, how to select stock index variables and preprocess data to remove noise is one of the key challenges in stock price prediction.

(2) Stock prices exhibit significant nonlinearity

The fluctuations in stock price time series and the factors influencing stock prices exhibit highly nonlinear characteristics. Therefore, a high-quality stock price prediction method should possess robust capabilities for handling nonlinear issues. Unfortunately, traditional prediction models are primarily designed to address linear problems, and they struggle to effectively handle the complex nonlinear nature of the stock market, resulting in suboptimal prediction outcomes that require further refinement.

(3) Stock investors exhibit subjective agency

The operational mechanisms of the stock market are highly complex. Stock prices are influenced not only by macro-level political and economic policies, meso-level industry and regional factors, and micro-level corporate operational conditions but also by the investment decisions of socially active investment groups with subjective agency. The investment decisions of market participants exhibit significant volatility and subjectivity. The uncertainty of psychological expectations and trading behavior impacts the accurate prediction of stock market price trends, necessitating a comprehensive consideration of these factors that may influence stock price fluctuations. Exploring the correlations among multiple factors to uncover the objective patterns underlying stock index fluctuations.

(4) Uncertainty in forecasting

The ongoing evolution and inherent uncertainty of the stock market determine the challenges in forecasting and analyzing it. Selecting appropriate stock market forecasting methods for analyzing and predicting individual stocks or the overall market is a hot topic of research among scholars worldwide. Models are fitted using sample data, and predictions are made for data outside the sample. Evaluating whether the selected model's performance meets standards, whether prediction errors are reduced, or whether prediction accuracy is improved, the core of the evaluation is to rank and compare the predictive capabilities of models based on an appropriate evaluation metric. However, in actual stock market prediction analysis, models may not perfectly capture the fluctuations of stock indices in reality. Therefore, the construction of the loss function is not the challenge, but rather how to establish an appropriate predictive model to achieve more precise and scientific prediction output values based on evaluation metrics.

(5) The uniqueness of the Chinese stock market

The development of China's securities market began relatively late, emerging in the 1980s and gradually taking shape by the end of the 20th century. China's stock market structure is quite unique, differing significantly from those of Western and European countries. Stock market volatility is high and cycles are irregular, especially in China's market, which is still in the process of development and improvement. Stock market changes lack overall coherence, with a high proportion of individual investors. However, differences in investor psychology, blind herd mentality, and the herd effect make it difficult to grasp stock price fluctuation patterns, thereby increasing the difficulty of prediction. Therefore, how to conduct more accurate stock market predictions has become a hotly debated topic in the economic field, as well as a highly challenging one.

II. B. BP Neural Network

II. B. 1) BP Network Structure

A BP neural network consists of three parts: an input layer, a hidden layer, and an output layer. The number of

hidden layers can be set to one or more layers. The structure of a BP neuron is shown in Figure 1.

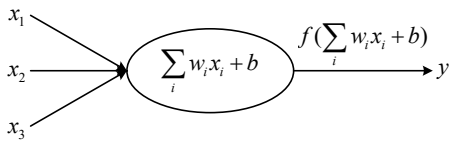


Figure 1: Structure diagram of BP neurons

Among these, x_1, x_2, x_3 are input nodes, y is the output node, w_1, w_2, w_3 are weight vectors, b is the threshold, and f represents the activation function [36]. During neural network training, the weight vectors w_i are continuously adjusted to minimize the error. At the same time, the activation function is correspondingly increased or decreased based on the positive or negative value of the threshold.

The number of layers in a BP neural network, as well as the number of nodes in its input, output, and hidden layers, may vary depending on the situation. The basic structure of a BP neural network is shown in Figure 2. The figure represents a three-layer BP neural network with three nodes in the input layer, five nodes in the hidden layer, and one output node. The figure shows the basic structure of a BP neural network, which is a three-layer neural network with a single hidden layer. For the structure of a BP neural network, the number of hidden layers can be increased according to actual requirements to form a more complex structure.

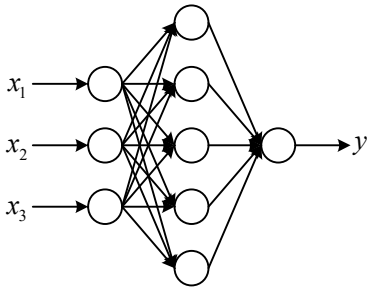


Figure 2: Basic Structure of BP Neural Network

II. B. 2) BP Network Algorithm

BP neural networks require training on data to continuously adjust the weights and thresholds within the network. During the training process, the weights and thresholds of each layer are adjusted according to the training algorithm. There are numerous training algorithms available, with the most common being gradient descent. When applying gradient descent, the gradient vector of the error with respect to the weights or thresholds is calculated, and the direction opposite to this gradient vector indicates the direction in which the weights and thresholds should be adjusted. However, gradient descent has certain limitations, leading to the development of various algorithms aimed at addressing these limitations. The LM algorithm essentially combines gradient descent with the Newton method. Gradient descent often converges quickly at the beginning of training, but when it approaches the optimal value, the gradient becomes very small, approaching zero, causing the convergence speed to slow down. The Newton method, however, can search for the optimal direction, enabling faster convergence to the optimal value. Combining these two methods forms the LM algorithm. The advantage of the LM algorithm is that it achieves faster convergence when the number of network parameters is small. However, since the LM algorithm involves the Hessian matrix, approximating it requires significant storage capacity. Therefore, the LM algorithm is most effective when the neural network is of medium or small scale.

II. B. 3) Design of the BP Network

(1) Determining the number of network layers

BP neural networks can be configured with multiple hidden layers, which typically need to be set according to specific requirements. However, existing theories have demonstrated that a BP network only requires a single hidden layer to implement any nonlinear mapping required by the neural network. When the number of samples is relatively small, a smaller number of hidden layer nodes is sufficient to meet the requirements. When the number of samples is large, an excessive number of hidden nodes can lead to a complex network structure, in which case an additional hidden layer should be added. Generally, the number of hidden layers in a BP neural network should not exceed two.

(2) Determining the Number of Neurons per Layer

While there is no definitive consensus on the optimal number of hidden layer nodes, the following empirical formula is widely used:

$$n = \sqrt{n_1 + n_0} + a \tag{1}$$

where n is the number of hidden layer nodes to be determined, n_1, n_0 are the numbers of input and output nodes, respectively. a is a constant between 1 and 10. This paper will also apply this method to determine the range of hidden node numbers.

(3) Activation functions for each layer

The role of the activation function is to transform the input and output values of each layer through a function, converting input variables that may have an infinite range into outputs within a limited range. The activation function of a BP neural network is nonlinear, monotonic, and differentiable. It can act on the input variables to perform nonlinear transformations. It is precisely this nonlinear transformation that allows the neural network to approximate any function without requiring an excessive number of hidden layers. Common activation functions include the logsig function and the tansig function, both of which are S-shaped functions. S-shaped functions can nonlinearly amplify coefficients, and any data range converted through them will have values between -1 and 1 [37]. In addition to S-shaped functions, pure linear functions are also used in the output layer. This is because the output results of the logsig and tansig functions are between -1 and 1, and the results need to be converted back to the original numerical range, while the output values of the purelin function can be any value.

II. B. 4) Training process of BP network

The training process for BP neural networks is as follows:

1. Initialize the data.
2. Determine the input variables and expected output results for the neural network, and train the input samples.
3. Obtain the output results and calculate the error based on the existing expected output.
4. Compare the actual error with the set error to see if the error size meets the requirements.
5. Continuously adjust the network weights and thresholds according to the algorithm until the error size meets the requirements.

II. B. 5) Defects in the BP Network

BP neural networks have some limitations in their application, which can be summarized as follows:

1. Requires a relatively long training time

For more complex problems, the training time required by the BP algorithm may be relatively long, which is related to the selection of an overly small learning rate. The gradient descent method typically sets a small learning rate to ensure stable learning, resulting in the neural network requiring a longer time to converge. For this issue, an adaptive learning rate or other algorithms such as the LM algorithm can be adopted.

2. Local minima

BP neural networks are prone to getting stuck in local minima during application. Optimization algorithms are typically used to address this issue. For example, the genetic algorithm introduced in this paper is an improvement for this problem.

3. The number of hidden layer neurons cannot be effectively determined

Currently, empirical formulas are commonly used to determine the range of hidden layer neuron counts, followed by trial-and-error selection to identify the optimal number of neurons.

II. C.Principal component analysis method

II. C. 1) Overview of Principal Component Analysis

Principal component analysis can be expressed by the following mathematical formula: Let $x_1, x_2, ..., x_p$ be p standardized random variables that are correlated with each other. Transform them linearly into one or more uncorrelated variables y_i, which are expressed as follows:

$$\begin{cases} y_1 = u_{11}x_1 + u_{12}x_2 + \ldots + u_{1p}x_p \\ y_2 = u_{21}x_1 + u_{22}x_2 + \cdots + u_{2p}x_p \\ \vdots \\ y_p = u_{p1}x_1 + u_{p2}x_2 + \ldots + u_{pp}x_p \end{cases} \tag{2}$$

Among them, $u_i 1^2 + u_{i2}^2 + \ldots + u_{ip}^2 = 1$. y_i is the i th principal component, and any principal components are uncorrelated. y_1 has the largest variance, followed by y_2, and so on. The above transformation can be expressed by the formula $\vec{Y} = U' \vec{X}$, where $\vec{U}$ is the principal component coefficient matrix.

II. C. 2) Basic steps of principal component analysis

The steps of principal component analysis are as follows:

1 Standardize the sample data. Apply the standardized data to subsequent analyses. Standardization allows the data to be on the same quantitative level and converts it into dimensionless data. There are many methods for standardizing data. This paper chooses the Z-score method to standardize the sample data X and convert it to A.

2 Calculate the correlation coefficient matrix R of the standardized A.

3. Based on the R obtained in the previous step, calculate its eigenvalues and eigenvectors. Solve the characteristic equation: $|R - \lambda_j E| = 0$, thereby obtaining the eigenvalues λ_j. Solve $(R - \lambda_j E)u_k = 0$, thereby obtaining the eigenvectors u_k.

4. The principal components can be obtained from the eigenvalues. Arrange the eigenvalues λ_j obtained in the previous step in descending order. Calculate the cumulative contribution rate of each principal component and select the first m principal components with a cumulative contribution rate greater than 85%.

5. Calculate the principal component loadings and scores. The principal component loading $l_{ij} = \sqrt{\lambda_i} u_{ij}$. The principal component score $\tau_i = a_i * u_i$.

II. D.PCA-BP neural network model

The PCA-BP model is a combination of principal component analysis and a neural network model. The original variables $x_1, x_2, \ldots, x_n$ are used to calculate the principal components $y_1, y_2, \ldots, y_p$ are calculated, and the principal component variables are used as input variables for the neural network. These are then input into the training model, and the final output results are obtained. The prediction process of the PCA-BP neural network is shown in Figure 3.

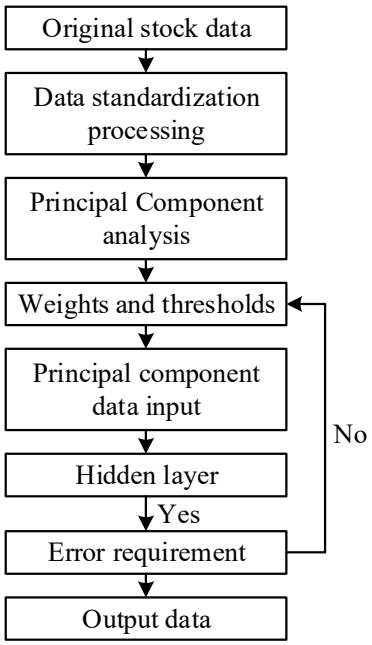


Figure 3: Prediction process of PCA-BP neural network

III. Experimental analysis

III. A. Model-based stock data analysis

III. A. 1) Experimental Data

The selection criteria for the CSI 300 Index are highly stringent, with stock size and liquidity serving as the

fundamental standards. This ensures a certain degree of stability for the index. As a trading-oriented component index, it continuously updates its constituent stocks due to factors such as a company's operational decisions or structural adjustments. Therefore, the CSI 300 Index fundamentally reflects the fluctuations in the Shanghai and Shenzhen stock markets. This paper uses the CSI 300 Index as its research data, with data sourced from the East Money software data download. The historical data of the CSI 300 Index used in this paper covers the period from March 1, 2010, to December 26, 2024. After excluding the influence of objective factors such as holidays, there are a total of 4,269 sets of stock prices, average prices, and trading volume data. Based on the selected variable calculation formulas, the final dataset consists of 4,000 rows and 40 columns. The dataset is divided into two parts: the first 3,800 rows are used for model training, and the remaining 200 rows are used for testing. The daily closing price charts for all sample data are shown in Figure 4. The study covers a broad time span, including government policies and economic crises, which generally indicates that the sample data is sufficient. The descriptive statistical results of the CSI 300 Index closing prices are shown in Table 1. As shown in the table, the skewness value of the CSI 300 Index closing prices is greater than 0, and the kurtosis value is less than 3, indicating that the closing prices do not follow a normal distribution.

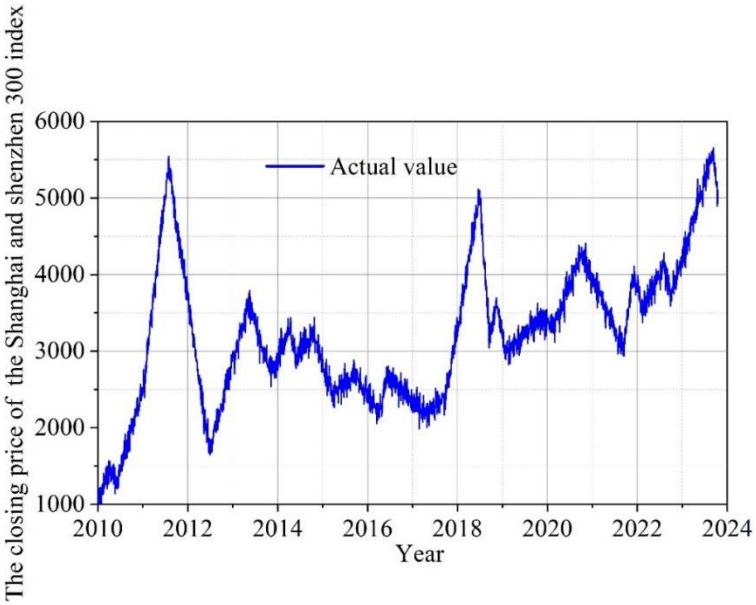


Figure 4: Daily chart of the closing price of the CSI 300 index

Table 1: Descriptive statistical results of the closing price of the CSI 300 Index

The CSI 300 Index	All samples	Training sample	Test sample
Sample size	4000	3800	200
Mean	2474.6	2449.8	3341.8
Standard deviation	1110.2	1101.7	89.15
Covariance	12374.6	12407.5	8045
Maximum value	5875.9	5875.3	3564.4
Minimum value	983.2	817.8	3177.8
Degree of bias	0.451	0.532	0.543
Kurtosis	2.711	2.772	2.745

Hidden insider information in the stock market is difficult to uncover solely based on price changes. At this point, it is necessary to utilize technical indicators of the stock market and comprehensively consider two factors in order to improve the accuracy of predictions. This paper selects 40 stock market variables that are currently receiving the most attention from researchers, with the closing price of the following day as the output variable. The variables are shown in Table 2.

Table 2: Table of influencing factor variables

Variable type	Symbol	Name
Independent variable	X1	Closing price
Independent variable	X2	Maximum price
Independent variable	X3	Lowest price
Independent variable	X4	Opening price
Independent variable	X5	The amount of the rise and fall
Independent variable	X6	Fluctuation
Independent variable	X7	Net volume index
Independent variable	X8	Volume variation rate
Independent variable	X9	Day before closing
Independent variable	X10	The day before
Independent variable	X11	Last day
Independent variable	X12	The previous day the opening price
Independent variable	X13	Popularity index
Independent variable	X14	Willingness indicator
Independent variable	X15	Momentum closing price
Independent variable	X16	Momentum maximum
Independent variable	X17	Momentum lowest price
Independent variable	X18	Momentum opening price
Independent variable	X19	Fast Stochastic indicator %K
Independent variable	X20	Fast Stochastic indicator %D
Independent variable	X21	Slow stochastic indicator %K
Independent variable	X22	Slow stochastic indicator %D
Independent variable	X23	The 5-day simple moving average of the closing price
Independent variable	X24	The 6-day simple moving average of the closing price
Independent variable	X25	The 10-day simple moving average of the closing price
Independent variable	X26	The 20-day simple moving average of the closing price
Independent variable	X27	The 5-day simple moving average of the closing price
Independent variable	X28	The 6-day simple moving average of the closing price
Independent variable	X29	The 10-day simple moving average of the closing price
Independent variable	X30	The 20-day simple moving average of the closing price
Independent variable	X31	The William Indicator on the 3rd W%R
Independent variable	X32	The 5-day William indicator W%R
Independent variable	X33	The 10-day William indicator W%R
Independent variable	X34	The Williams indicator W%R on the 21st
Independent variable	X35	Moving average summary
Independent variable	X36	The daily index moves the average line
Independent variable	X37	Relative strength index
Independent variable	X38	Deviation rate indicator
Independent variable	X39	Price variation
Independent variable	X40	Psychological line
Dependent variable	Y	The closing price of the CSI 300 Index on the second day

III. A. 2) Experimental results of PCA-BP neural network

① Principal Component Analysis (PCA)

First, create a 3800 x 40 matrix from the original data samples. Due to the large amount of data, standardize it first using MATLAB 2018a, and use A1 to represent the processed data matrix. Step 2: For the processed data, the princomp function in MATLAB is called to calculate the parameters corresponding to the principal component analysis. Some of the parameters after PCA principal component analysis are shown in Table 3.

Table 3: Some Parameters after the PCA principal component analysis

Principal component	Eigenvalue	Eigencontribution (%)	Cumulative contribution rate of features (%)
1	27.31	58.96	58.96
2	10.52	25.63	84.59
3	4.184	6.321	90.911
4	1.527	3.856	94.767
5	1.104	1.856	96.623
6	0.865	0.865	97.488
……	……	……	……

② Selection of the number of principal components

The variables in the input layer of the neural network are the principal components we need. Currently, the number of principal components is determined based on the cumulative contribution rate of features and the criterion that Cvca ≥ 85%. According to the results in the table, the cumulative contribution rate of the first three principal component variables is over 90%. Therefore, only three principal components are needed to ensure that the majority of the data features are represented, and the selected data has a certain degree of representativeness. Different numbers of principal components have different effects on BP neural network predictions, so we need to verify whether the three principal components mentioned above have the optimal effect on BP neural network predictions. In this section, we determine the number of principal components based on the table. As shown in the table, when the number of principal components increases to six, the feature contribution rate decreases to less than 1%, and the effect is almost negligible. Therefore, we select integers between 1 and 5 as the number of principal components for the experiment. The learning rate is set to 0.1, the accuracy to 1e−4, and the number of training iterations to 100. The experimental results are the average of 100 experiments, and the standard deviation of the evaluation metrics is calculated to obtain the following experimental results. The test results for the number of principal components of the CSI 300 Index are shown in Table 4. From the table, it can be seen that the optimal number of principal components is 2, yielding the best results.

Table 4: Test results of the number of principal components of the CSI 300 Index

Principal component		MAE	MSE	RMSE	MARE	MSRE	RMSRE	MAPE	MSPE	RMSPE
1	Mean	29.54	1532.1	38.08	0.007	0.001	0.015	0.885	0.016	0.1154
	Standard deviation	0.363	25.24	0.325	9.741	1.816	7.874	0.008	0.0002	0.0007
2	Mean	22.81	912.2	20.20	0.005	8.2E-05	0.009	0.678	0.0080	0.0913
	Standard deviation	0.214	12.63	0.209	6.514	1.195	6.526	0.003	0.0001	0.0008
3	Mean	22.24	941.5	30.75	0.007	8.5E-05	0.009	0.687	0.0086	0.0944
	Standard deviation	0.459	38.97	0.655	0.001	3.6E-06	0.001	0.014	0.0004	0.0025
4	Mean	24.21	1035.1	32.45	0.006	9.2E-05	0.008	0.716	0.0092	0.0985
	Standard deviation	1.124	100.5	1.479	0.001	9.1E-06	0.001	0.036	0.0009	0.0038
5	Mean	22.24	853.72	29.23	0.006	7.5E-05	0.007	0.668	0.0086	0.0871
	Standard deviation	0.913	62.75	1.026	0.001	5.5E-06	0.001	0.027	0.0005	0.0053

After performing principal component analysis (PCA) on the data using MATLAB, it can be concluded that when the number of principal components is set to 2, the prediction error is minimized for each training set of 100. By using the variables corresponding to the principal components, a representative analysis can be conducted. The corresponding principal component coefficient matrices are shown in Table 5. As shown in the table, the maximum value of the first principal component coefficient matrix is 0.199, and the maximum value of the second principal component coefficient matrix is 0.283. By comparing the values corresponding to each variable in the first principal component coefficient matrix and the second principal component coefficient matrix, we determine that variables with values greater than 0.19 in the first principal component coefficient matrix are major influencing factors, and variables with values greater than 0.2 in the second principal component coefficient matrix are major influencing factors. Therefore, the first principal component includes 23 variables, the second principal component includes 15 variables, and a total of 38 variables are selected.

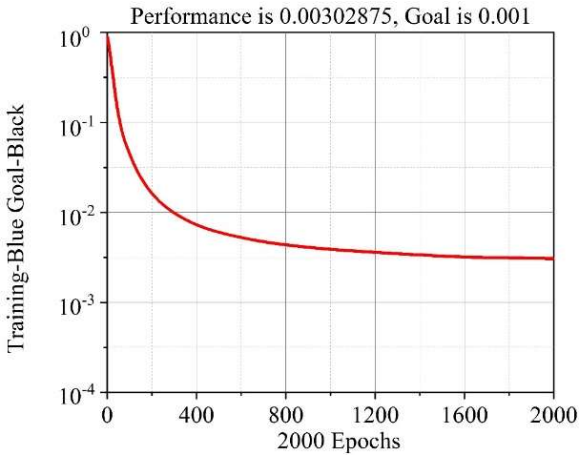
Table 5: Principal component coefficient Matrix

Variable Type	The first principal component	The second principal component
X1	0.199	0.001
X2	0.193	-0.002
X3	0.197	0
X4	0.196	-0.002
X5	0.195	-0.002
X6	0.003	0.094
X7	0.193	-0.002
X8	0.196	-0.001
X9	0.195	-0.003
X10	0.194	-0.005
X11	0.198	-0.006
X12	0.192	-0.006
X13	0.029	0.223
X14	-0.006	0.201
X15	0.008	0.276
X16	0.016	0.277
X17	0.015	0.28
X18	0.014	0.283
X19	0.025	0.245
X20	0.029	0.252
X21	0.028	0.264
X22	0.033	0.256
X23	0.192	-0.01
X24	0.195	-0.007
X25	0.192	-0.018
X26	0.197	-0.035
X27	0.193	-0.007
X28	0.198	-0.008
X29	0.191	-0.018
X30	0.196	-0.03
X31	0.194	-0.019
X32	0.193	-0.025
X33	0.196	-0.039
X34	0.193	-0.055
X35	0.023	0.259
X36	0.029	0.209
X37	-0.001	0.013
X38	0.009	0.266
X39	0.015	0.282
X40	0.035	0.232

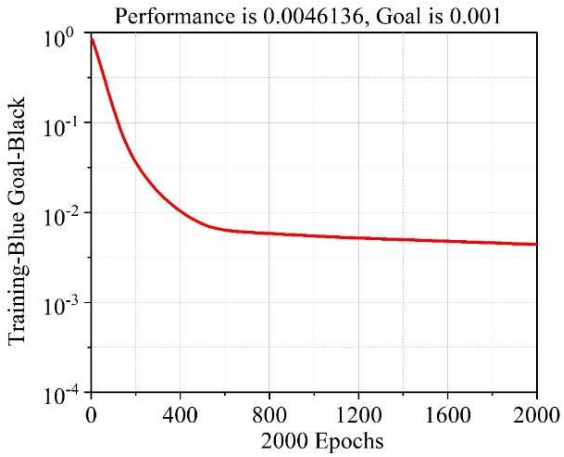
III. B. Model Comparison Analysis

The experimental method employed was as follows: all parameters were kept consistent except for the number of input layer variables. A network structure with 11 hidden layer neurons and 1 output layer variable was first selected. The model training results are shown in Figure 5 (Figure a shows the PCA-BP training results, and Figure b shows the BP neural network training results). When the number of hidden layer neurons is 11, observing the figure reveals that during model training, compared to the initial 200 steps in Figures a and b, the error of PCA-BP decreases more rapidly than that of the BP neural network, entering a smoother state sooner. This indicates that the PCA-BP training speed is faster than that of the BP neural network. In subsequent training, the PCA-BP training error is smaller than that of the BP neural network, indicating that the prediction accuracy of the PCA-BP is higher than that of the BP neural network. The output simulation is shown in Figure 6 (Figure a shows the true values and predicted values of the PCA-BP, and Figure b shows the true values and predicted values of the BP neural network). The

points representing the actual values and predicted values of the PCA-BP and BP neural networks are largely overlapping, indicating that their prediction error effects are essentially similar, with no significant differences. To further investigate the impact of different hidden layers on prediction results, additional simulation experiments were conducted by varying the number of neurons in the hidden layers of the BP neural network.

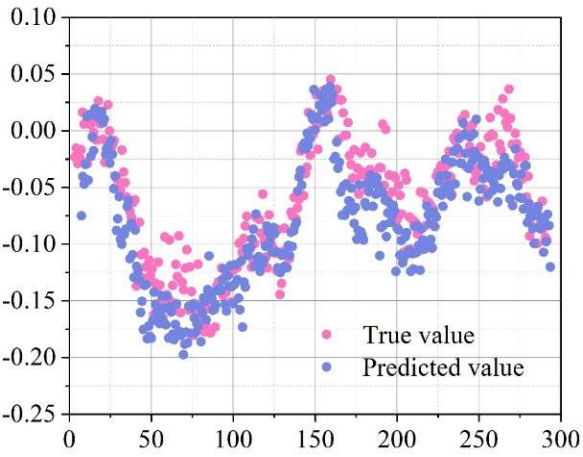


(a) PCA-BP training effect diagram

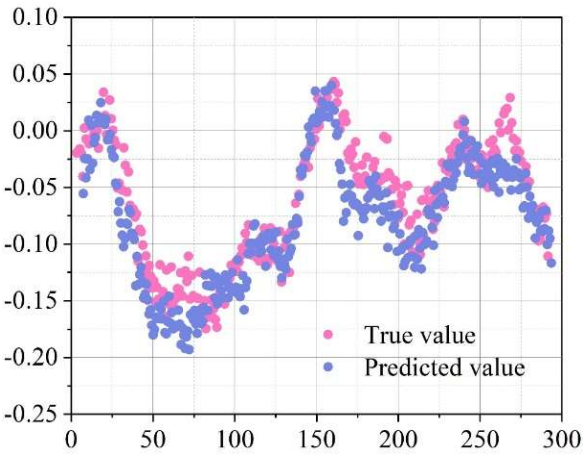


(b) BP neural network training rendering

Figure 5: The training effect of the input vector group



(a) PCA-BP true value and predicted value graph



(b) The graph of true values and predicted values of BP neural network

Figure 6: Input vector group output simulation

The MSE of PCA-BP and BP neural networks for the number of neurons in each hidden layer is shown in Table 6. As can be seen from the table, although both groups exhibit fluctuations in accuracy when the number of neurons in the hidden layer is selected differently, the points representing the MSE values of PCA-BP approximate a straight line, while the points representing the BP neural network exhibit greater variability. Furthermore, the MSE values of PCA-BP are significantly smaller than those of the BP neural network. This indicates that PCA-BP outperforms the BP neural network in terms of both accuracy and stability. When comparing the prediction results of the two groups with different input vector sets, we find that the most significant difference lies in the fact that the relationships between variables in PCA-BP are more reasonable and clear, whereas the relationships between variables in the BP neural network are complex and chaotic, which affects the prediction performance of the neural network. By analyzing the stock price prediction results of the BP neural network, the following basic conclusions can be drawn: when the information content of the data is the same, the number of neurons in the hidden layer of the network affects the prediction accuracy. The PCA-BP, with its superior information structure, not only improves prediction accuracy but also enhances its training speed and stability.

Table 6: The MSE of the number of neurons in each hidden layer

Input vector	Main component vector group (6)	Improved Principal Component Vector Group (6)
MSE(6)	0.002662033272322	0.000126236765222
MSE(7)	0.003263670360073	0.000300622063302
MSE(8)	0.002303323027633	0.000232226625222
MSE(9)	0.001663602512622	0.000233700470767
MSE(10)	0.003677230732736	0.000222676277332
MSE(11)	0.001223270346232	0.000422602132603
MSE(12)	0.002322267760632	0.000302317663022
MSE(13)	0.001772630327022	0.000123325202366
MSE(14)	0.003323620622263	0.000273233266763
MSE(15)	0.001602626236026	0.000162377267622

III. C. Accuracy of model predictions

III. C. 1) Direction accuracy rate

If a stock's closing price on a given day is higher than the previous day's closing price, and the predicted closing price is also higher than the previous day's closing price, then we can consider this a correct prediction without needing to compare the actual closing price with the predicted closing price. The direction of actual and predicted price changes is shown in Table 7, with “+” indicating an increase and “-” indicating a decrease. As shown in the table, out of 13 predictions, 11 were correct, resulting in a directional accuracy rate of 85.63%. This indicates that, from the perspective of directional accuracy, the predictive performance of the comprehensive model is still quite good.

Table 7: Direction accuracy rate

Stock code	Actual rise and fall	Predict the rise and fall
600232.SH	-	-
600143.SH	-	-
600867.SH	-	-
002214.SZ	-	-
600134.SH	-	-
300246.SZ	-	-
600567.SH	-	-
600367.SH	-	-
600434.SH	-	-
600735.SH	+	-
600961.SH	-	-
600243.SH	-	-
000778.SZ	-	-
Direction accuracy rate	92.31%	

III. C. 2) Data Validation

To further validate the model's generalization ability, this paper uses transaction data and financial data from July 30 to predict the closing price on August 2 from 13 samples. The comparison between the predicted values and the actual values is shown in Table 8. Observing the mean squared error and the mean percentage error of the model's predictions, it is found that the prediction accuracy is not ideal. This is primarily because the predicted time point is too far from the time point at which the training samples were selected, and the data structure and properties may have undergone significant changes during this period. Additionally, this paper primarily selected cross-sectional data, but for time series data predictions, time series data should actually be selected to ensure continuity. However, since financial data remains unchanged over a certain period, and there are two methods for model validation, if time series data is selected, the model can be validated using data from a subsequent period.

The accuracy rate of the direction on December 5 is shown in Table 9. Out of 13 predictions, 12 were correct, with an accuracy rate of 92.31%, which provides some guidance for us in determining whether stock prices will rise or fall.

Table 8: Compare the predicted value with the actual value

Stock code	Actual value	Predicted value	Absolute error	Relative error
002142.SZ	9.6776	9.0487	0.6289	0.0650
6001298.SH	14.3051	11.4872	2.8179	0.1970
000733.SZ	28.123	21.8331	6.2899	0.2237
300022.SZ	73.9898	72.3659	1.6239	0.0219
600116.SH	16.4537	13.1706	3.2831	0.1995
002289.SZ	15.7831	15.2709	0.5122	0.0325
600302.SH	19.0732	13.7282	5.345	0.2802
600746.SH	16.3399	15.4558	0.8841	0.0541
002301.SZ	10.1646	9.85	0.3146	0.0310
000693.SZ	17.2079	14.2111	2.9968	0.1742
002185.SZ	24.5155	21.5902	2.9253	0.1193
603669.SH	37.6168	28.306	9.3108	0.2475
000921.SZ	19.0685	14.7641	4.3044	0.2257
MSE			11.8591	
MAPE				0.1332

Table 9: The direction accuracy rate on December 5th

Stock code	Actual rise and fall	Predict the rise and fall
002164.SZ	-	-
600126.SH	-	-
000773.SZ	+	-
300011.SZ	-	-
600432.SH	-	-
002282.SZ	-	-
600163.SH	-	-
600360.SH	-	-
002301.SZ	-	-
000631.SZ	-	-
002185.SZ	-	-
603674.SH	-	-
000836.SZ	-	-
Direction accuracy rate	92.31%	

IV. Conclusion

The stock market plays a crucial role in the development of the modern economy. A healthy and well-regulated stock market can facilitate capital flow, optimize resource allocation, and enhance the efficiency of capital utilization. This paper selects data from the CSI 300 Index and employs a PCA-BP neural network model for experimentation, yielding the following results:

In the input vector group output simulation experiment, the improved principal component vector group and the principal component vector group's actual values and predicted values are largely overlapping, indicating that their prediction error effects are essentially comparable.

In the directional accuracy experiment on December 5, the model correctly predicted 12 out of 13 instances, achieving an accuracy rate of 92.31%. This validates the model's excellent predictive performance, which provides valuable guidance for assessing stock price fluctuations.

References

[1] Kotishwar, A. (2020). The Impact of Currency Fluctuations on Equity and Debt Market. International Journal of Economics and Business Administration, 8(4), 392-406.

[2] Anand, K., Khedair, J., & Kühn, R. (2018). Structural model for fluctuations in financial markets. Physical Review E, 97(5), 052312.

[3] Akter, R. (2021). Factors that Determine and Influences Foreign Exchange Rates. Social Islamic Bank Limited.

[4] Ahmed, W. M. (2017). The impact of political regime changes on stock prices: the case of Egypt. International Journal of Emerging Markets, 12(3), 508-531.

[5] Qian, Y., Ralescu, D. A., & Zhang, B. (2019). The analysis of factors affecting global gold price. Resources Policy, 64, 101478.
[6] Ammer, M. A., & Aldhyani, T. H. (2022). Deep learning algorithm to predict cryptocurrency fluctuation prices: Increasing investment awareness. Electronics, 11(15), 2349.
[7] Idrees, S. M., Alam, M. A., & Agarwal, P. (2019). A prediction approach for stock market volatility based on time series data. IEEE Access, 7, 17287-17298.
[8] Fabretti, A. (2022). A dynamical model for financial market: Among common market strategies who and how moves the price to fluctuate, inflate, and burst?. Mathematics, 10(5), 679.
[9] Lin, Y. F., Huang, T. M., Chung, W. H., & Ueng, Y. L. (2020). Forecasting fluctuations in the financial index using a recurrent neural network based on price features. IEEE Transactions on Emerging Topics in Computational Intelligence, 5(5), 780-791.
[10] Wang, G., Zheng, S., & Wang, J. (2021). Nonlinear fluctuation behaviors of complex voter financial price dynamics on small-world network. Nonlinear Dynamics, 103(3), 2525-2545.
[11] Jiang, L. C., & Subramanian, P. (2019). Forecasting of stock price using autoregressive integrated moving average model. Journal of Computational and Theoretical Nanoscience, 16(8), 3519-3524.
[12] Lin, Z. (2018). Modelling and forecasting the stock market volatility of SSE Composite Index using GARCH models. Future Generation Computer Systems, 79, 960-972.
[13] Alshammari, T. T., Ismail, M. T., Hamadneh, N. N., Al Wadi, S., Jaber, J. J., Alshammari, N., & Saleh, M. H. (2023). Forecasting Stock Volatility Using Wavelet-based Exponential Generalized Autoregressive Conditional Heteroscedasticity Methods. Intelligent Automation Soft Computing, 35, 2589-601.
[14] Abdullah, S. M., Siddiqua, S., Siddiquee, M. S. H., & Hossain, N. (2017). Modeling and forecasting exchange rate volatility in Bangladesh using GARCH models: a comparison based on normal and Student'st-error distribution. Financial Innovation, 3, 1-19.
[15] Özorhan, M. O., Toroslu, İ. H., & Şehitoğlu, O. T. (2017). A strength-biased prediction model for forecasting exchange rates using support vector machines and genetic algorithms. Soft Computing, 21, 6653-6671.
[16] Nadh, V. L., & Prasad, G. S. (2018). Support vector machine in the anticipation of currency markets. Int. J. Eng. Technol, 7(2-7), 66.
[17] Mohammed, S. A. S. A. (2025). Support Vector Machines in Stock Price Prediction: A Review. Financial Innovation for Global Sustainability, 561-578.
[18] Zhang, D., & Lou, S. (2021). The application research of neural network and BP algorithm in stock price pattern classification and prediction. Future Generation Computer Systems, 115, 872-879.
[19] Alameer, Z., Abd Elaziz, M., Ewees, A. A., Ye, H., & Jianhua, Z. (2019). Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm. Resources Policy, 61, 250-260.
[20] Vrtagic, S., & Dogan, F. (2024). High-Frequency Gold Price Forecasting: Optimizing Multi-Layer Perceptron with Genetic Algorithm. Mathematical Modelling of Engineering Problems, 11(12).
[21] Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A deep learning-based cryptocurrency price prediction scheme for financial institutions. Journal of information security and applications, 55, 102583.
[22] Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. Intelligent Systems in Accounting, Finance and Management, 26(4), 164-174.
[23] Pan, H., Tang, Y., & Wang, G. (2024). A Stock Index Futures Price Prediction Approach Based on the MULTI-GARCH-LSTM Mixed Model. Mathematics, 12(11), 1677.
[24] Zhang, X., Zhang, L., Zhou, Q., & Jin, X. (2022). A novel Bitcoin and Gold prices prediction method using an LSTM-P neural network model. Computational Intelligence and Neuroscience, 2022(1), 1643413.
[25] Dey, P., Hossain, E., Hossain, M. I., Chowdhury, M. A., Alam, M. S., Hossain, M. S., & Andersson, K. (2021). Comparative analysis of recurrent neural networks in stock price prediction for different frequency domains. Algorithms, 14(8), 251.
[26] Li, M., & Wang, Z. (2020). Deep learning for high-dimensional reliability analysis. Mechanical Systems and Signal Processing, 139, 106399.
[27] Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2017, August). Using deep learning to detect price change indications in financial markets. In 2017 25th European signal processing conference (EUSIPCO) (pp. 2511-2515). IEEE.
[28] Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Applied Soft Computing, 70, 525-538.
[29] Wu, J., Xu, K., Chen, X., Li, S., & Zhao, J. (2022). Price graphs: Utilizing the structural information of financial time series for stock prediction. Information Sciences, 588, 405-424.
[30] Wu, J. M. T., Li, Z., Srivastava, G., Tasi, M. H., & Lin, J. C. W. (2021). A graph-based convolutional neural network stock price prediction with leading indicators. Software: Practice and Experience, 51(3), 628-644.
[31] Gewers, F. L., Ferreira, G. R., Arruda, H. F. D., Silva, F. N., Comin, C. H., Amancio, D. R., & Costa, L. D. F. (2021). Principal component analysis: A natural approach to data exploration. ACM Computing Surveys (CSUR), 54(4), 1-34.
[32] Rodionova, O., Kucheryavskiy, S., & Pomerantsev, A. (2021). Efficient tools for principal component analysis of complex data—A tutorial. Chemometrics and Intelligent Laboratory Systems, 213, 104304.
[33] Manoj, J., & Suresh, K. K. (2019). Forecast model for price of gold: Multiple linear regression with principal component analysis. Thailand Statistician, 17(1), 125-131.
[34] Zhang, Z. (2022). Research on stock price prediction based on PCA-LSTM model. Academic Journal of Business & Management, 4(3), 42-47.
[35] Zhang, H. (2018). The forecasting model of stock price based on PCA and BP neural network. Journal of Financial Risk Management, 7(04), 369.
[36] LeiHang,DandanLiu & FushengXie. (2023). A Hybrid Model Using PCA and BP Neural Network for Time Series Prediction in Chinese Stock Market with TOPSIS Analysis. Scientific Programming,2023(1),9963940-9963940.
[37] Sheng Yankai,Fu Kui & Liang Jing. (2022). Construction of a Fundamental Quantitative Evaluation Model of the A-Share Listed Companies Based on the BP Neural Network. Computational Intelligence and Neuroscience,2022,7069788-7069788.